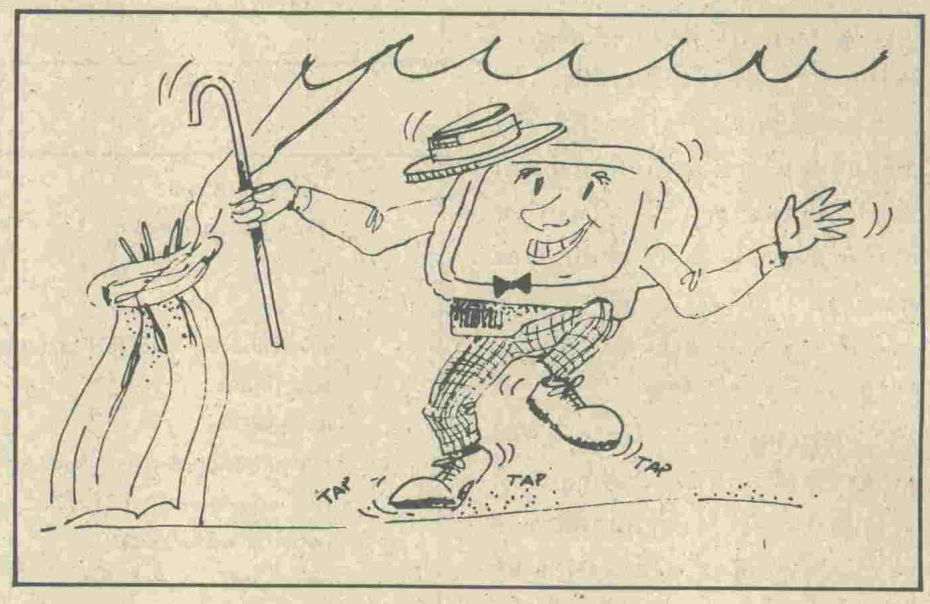
อาจหาญ สัตยารักษ์

SOFT MAIL

คอมพิวเตอร์ร้องเพลงได้อย่างไร



เปิดเผยแนวทางในการที่จะทำให้คอมพิวเตอร์ร้องเพลงได้อย่างสมบูรณ์แบบ

ก่อนอื่นผู้เขียนขอเกริ่นให้ทราบว่าบทความนี้เหมาะกับบุคคล 2 กลุ่ม คือ ผู้เขียนโปรแกรมด้วยเทอร์โบปาสคาลกลุ่มหนึ่ง และผู้ใช้โปรแกรมกราฟิก THAISHOW รุ่น 3.0 อีกกลุ่มหนึ่ง

เหตุผลที่ผู้เขียนกล่าวอย่างนี้เพราะ

1. บทความนี้เกิดจากผู้ใช้โปรแกรมกราฟิก THAISHOW ที่ให้ข้อคิดเห็นในเรื่องการสร้างเสียง

2. บทความนี้ได้มีลิสติงที่เขียนด้วยเทอร์โบปาสคาล เพื่อเผยแพร่ให้การใช้งานด้านเสียงมีการใช้งานที่กว้างขวางมากขึ้น

ผู้เขียนได้จัดทำเป็นซอฟต์เมล์เช่นเดิม โดยมีไฟล์ทั้งสิ้น 3 ไฟล์ดังนี้

1. ไฟล์ SONG.PAS ไฟล์เริ่มต้นในการเปล่งเสียง

2. ไฟล์ HEAD.PAS ไฟล์ที่เป็นส่วนหัวของไฟล์ song2.pas

3. ไฟล์ SONG2.PAS ไฟล์ในการให้โปรแกรมวนรอบเปล่งเสียงในช่วงการรอกดแป้น

ปฐมเหตุการเขียนบทความ

หลังจากที่โปรแกรมกราฟิก THAISHOW รุ่น 3.0 ได้วางตลาดไปเมื่อเดือนมีนาคม 2536 ปรากฏว่ามีผู้สนใจนำไปใช้งานต่าง ๆ มาก ทั้งยังช่วยอนุเคราะห์ให้คำแนะนำในเรื่องต่าง ๆ มากมาย ซึ่งผู้เขียนขอขอบคุณทุก ๆ ท่านมาพร้อมนี้ด้วย

ในส่วนของการกำหนดเสียง อาจารย์ขจรยศ เฉลิมญาติ จากโรงเรียนโคราชพิทยาคม อำเภอเมือง จังหวัดนครราชสีมา 30000 ได้ให้คำแนะนำที่เป็นประโยชน์มากดังนี้

ตามหลักโน้ตสากลกำหนดให้ใช้ตัวอักษรภาษาอังกฤษแทนเสียงต่าง ๆ ดังนี้

- C ใช้แทนเสียง โด
- D ใช้แทนเสียง เร
- E ใช้แทนเสียง มี
- F ใช้แทนเสียง ฟา
- G ใช้แทนเสียง ซอล
- A ใช้แทนเสียง ลา
- B ใช้แทนเสียง ที

โดยมีการกำหนดระดับเสียงที่เป็นครึ่งเสียง โดยการใช้เครื่องหมายดังนี้

(ชาร์ป) หมายถึง ทำให้โน้ตตัวนั้นมีระดับเสียงสูงขึ้นอีกครึ่งเสียง

b (แฟลต) หมายถึง ทำให้โน้ตตัวนั้นมีระดับเสียงต่ำลงอีกครึ่งเสียง

เครื่องหมายดังกล่าวจะเขียนไว้ข้างหลังมุมขวาบนของตัวอักษร เช่น $C^{\#}$ เรียกว่า โดชาร์ป

โครงสร้างบันไดเสียงเมเจอร์ (เดียโตนิค) จะมี 7 เสียง 8 ขั้น ดังแสดงในรูปที่ 1 ซึ่งช่วงเสียงใดมีระยะห่าง 1 เสียงเต็ม ก็จะสามารถแบ่งครึ่งได้โดยใช้เครื่องหมาย # หรือ b เช่น ถ้านับ

จากเสียง C ขึ้นไปอีกครึ่งเสียง (ระหว่างเสียง C กับ D) ก็จะเป็น เสียง C# (โดชาร์ป) แต่ถ้านับจาก D ลงมาครึ่งเสียง (ระหว่างเสียง C กับ D) ซึ่งเป็นจุดเดียวกัน จะได้เสียง Db (เรแฟลต) ทั้งนี้ขึ้นอยู่กับบันไดเสียงที่ใช้

แต่ในการเขียนโปรแกรมอ.ขจรยศมีความเห็นว่า ใช้เครื่องสำหรับกำหนดเสียงเพียงตัวใดตัวหนึ่งก็พอ (ที่นิยมใช้จะเป็นเครื่องหมาย #)

เสียงดนตรีประกอบด้วยส่วนสำคัญคือ

1. ระดับเสียง เช่น C D E F G A B C

2. ความหมายของเสียง (ค่าหน่วงเวลา) เช่น ตัวกลม ตัวขาว ตัวดำ ตัวเขบ็ต 1 ชั้น ตัวเขบ็ต 2 ชั้น

- ตัวกลม มีความยาวของเสียงเป็น 2 เท่าของตัวขาว
- ตัวขาว มีความยาวของเสียงเป็น 2 เท่าของตัวดำ
- ตัวดำ มีความยาวของเสียงเป็น 2 เท่าของตัวเขบ็ต 1 ชั้น
- ตัวเขบ็ต 1 ชั้น มีความยาวของเสียงเป็น 2 เท่าของตัวเขบ็ต 2 ชั้น

ถ้าสมมติให้ตัวกลมมีค่าเป็น 4 ตัวขาวจะเป็น 2 ตัวดำเป็น 1 เขบ็ต 1 ชั้นเป็น 1/2 เป็นต้น แต่ถ้าต้องการโน้ตที่มีความยาวเป็น 3 จะต้องนำโน้ตตัวขาวมาใส่จุดข้างหลัง (ค่อนข้างยุ่งยาก) ในทางปฏิบัติจะให้เลขตัวกลมเป็น 1, ตัวขาว 2, ตัวดำ 4, ตัวเขบ็ต 1 ชั้น 8, ตัวเขบ็ต 2 ชั้น 16, และให้ตัวขาวประจุดเป็น 3

ค่ามาตรฐาน 1 มีเสียงยาวที่สุด
2 ยาวครึ่งหนึ่งของ 1
4 ยาวครึ่งหนึ่งของ 2
8 ยาวครึ่งหนึ่งของ 4
16 ยาวครึ่งหนึ่งของ 8

รูปที่ 1 โครงสร้างบันไดเสียงเมเจอร์ (เดียโตนิค)

ในส่วนการกำหนดความยาวเสียงนี้อ.ขจรยศเห็นว่าควรใช้การกำหนดแบบเดิมจะเหมาะสมที่สุด คือเลขยิ่งมากระยะการเกิดเสียงยิ่งนาน เพราะจะสะดวกกับผู้ใช้ที่ไม่คุ้นกับระบบดนตรี

3. การพักเสียง (การหยุดเสียง)

4. OCTAVE ช่วงเสียง (คู่แปด)

5. ความเร็ว TEMPO

อ.ขจรยศมีข้อคิดเห็นดังนี้

ส่วนที่ 1 บอกความเร็ว เทียบจากนาฬิกา ถ้า TEMPO = 60 หมายถึงใน 1 นาที จะมี 60 จังหวะเคาะ ถ้าเทียบกับสัญญาณนาฬิกาในคอมพิวเตอร์ความเร็วเท่านี้ (60) 18 ติ๊ก จะมีค่าเท่ากับ 1 จังหวะเคาะ เพราะนาฬิกาในคอมพิวเตอร์จะเดิน 18 ติ๊ก/1 วินาที ฉะนั้นถ้าส่วนกำหนดความเร็วมากขึ้นเพลงต้องเร็วขึ้นด้วย

ส่วนที่ 2 เมื่อเริ่มเขียนโน้ตควรกำหนด OCTAVE ก่อน เช่น 30 (สามโอ) หมายถึงการเริ่มโน้ตใน OCTAVE ที่ 3 และหลังจากนั้นไม่ต้องกำหนดใหม่ ยกเว้นมีการเปลี่ยน OCTAVE

ส่วนที่ 3 บอกระดับเสียงเป็นอักษรแทนเสียงตามหลักสากล

ส่วนที่ 4 บอกค่าความยาวของเสียง (ค่าหน่วงเวลา)

ส่วนที่ 5 กำหนดให้งดการออกเสียงได้ชั่วคราวตามค่าหน่วงเวลาเดียวกับตัวโน้ต (ให้ออกเสียงที่เราไม่ได้ยินในช่วงความถี่ประมาณ 32767 เฮิรตซ์ขึ้นไป) และอาจใช้ตัวแปร P (Pause) นำหน้าความยาวของเสียงเงียบนี้

ส่วนที่ 6 โน้ตเสียงเดียวกัน เขียนติดกัน สามารถออกเสียงได้เป็นคนละตัว (ไม่ยาวติดต่อกัน) ขอให้สังเกตจากเพลงที่ผมเขียนมา เช่น โน้ตเพลงชาติไทย จะเห็นว่าบางท่อนมีโน้ตเสียงเดียวกันอยู่ติดกัน เวลาเล่นออกมาจะกลายเป็นเสียงเดียว แต่จุดประสงค์ต้องการเล่นโน้ตตัวนั้นซ้ำ ตัวอย่าง

```
120
30C4,D2,E1,F4,G8,40D2,P2,30A4,
```

เมื่อ 120 คือ ค่าความเร็วตามที่กำหนด (แต่ตัวเลขยิ่งมากจังหวะต้องเร็วขึ้น)

30 คือ กำหนดให้เล่นโน้ตใน OCTAVE ที่ 3

C4 คือ ให้เล่นโน้ตเสียง โด มีค่าความยาว 4

D2 คือ ให้เล่นโน้ตเสียง เร มีค่าความยาว 2

E1 คือ ให้เล่นโน้ตเสียง มี มีค่าความยาว 1

G8 คือ ให้เล่นโน้ตเสียง ซอล มีค่าความยาว 8

ทั้งหมดนี้ เล่นอยู่ใน OCTAVE 3

40D2 เล่น OCTAVE 4 โน้ตเสียง เร ยาว 2 P2 ให้หยุดการออกเสียง (หรือออกเสียงที่เราไม่สามารถได้ยิน) มีค่าความยาว 2 30A4 กลับมาเล่นโน้ต OCTAVE 3 เสียง ลา ยาว 4

แนวคิดนี้ผู้ใช้สามารถนำโน้ตที่วางขายตามท้องตลาด (โน้ตอิเล็กโทนที่เป็นตัวเลขและตัวอักษร) มาเขียนโน้ตเสียงเลย

ตารางตัวโน้ตและความถี่

สำหรับความถี่ของเสียงตามตัวโน้ตที่ใช้จะแยกตามช่วงคู่แปดมีค่าดังตารางที่ 1

การกำหนดระดับเสียง

เมื่อพิจารณาตามคำแนะนำของ อ.ขจรยศแล้ว ผู้เขียนจึงได้ปรับปรุงการเปล่งเสียงดังนี้

1. กำหนดให้ระดับเสียงมีทั้งหมด 13 ระดับ คือ C C# D D# E F F# G G# A A# B P โดยตัวอักษรภาษาอังกฤษแทนเสียงต่าง ๆ ดังนี้

- C ใช้แทนเสียง โด
- D ใช้แทนเสียง เร
- E ใช้แทนเสียง มี
- F ใช้แทนเสียง ฟา
- G ใช้แทนเสียง ซอล
- A ใช้แทนเสียง ลา
- B ใช้แทนเสียง ที
- # ใช้แทนเสียงครึ่งเสียง (เสียงชาร์ป)
- P ใช้แทนการพักเสียง

2. ให้ตัวเลขตามหลังตัวโน้ต (ในข้อ 1) เพื่อกำหนดความยาวของเสียง โดยตัวเลขน้อยเสียงจะสั้น ตัวเลขมากเสียงจะยาว

ตารางที่ 1
ตารางแสดงตัวโน้ตและความถี่

	โน้ต	ความถี่
Octave 1	C	131
	C#	139
	D	147
	D#	156
	E	165
	F	175
	F#	185
	G	196
	G#	208
	A	220
	A#	233
	B	247

	โน้ต	ความถี่
Octave 2	C	262
	C#	277
	D	294
	D#	311
	E	330
	F	349
	F#	370
	G	392
	G#	415
	A	440
	A#	466
	B	494

	โน้ต	ความถี่
Octave 3	C	523
	C#	554
	D	587
	D#	622
	E	659
	F	698
	F#	740
	G	784
	G#	831
	A	880
	A#	932
	B	988

	โน้ต	ความถี่
Octave 4	C	1047
	C#	1109
	D	1175
	D#	1245
	E	1319
	F	1397
	F#	1480
	G	1568
	G#	1661
	A	1760
	A#	1865
	B	1976

	โน้ต	ความถี่
Octave 5	C	2093
	C#	2217
	D	2349
	D#	2489
	E	2637
	F	2794
	F#	2960
	G	3136
	G#	3322
	A	3520
	A#	3729
	B	3951

	โน้ต	ความถี่
Octave 6	C	4186
	C#9	4435
	D	4699
	D#	4978
	E	5274
	F	5587
	F#	5919
	G	6271
	G#	6645
	A	7040
	A#9	7459
	B	7902

3. กำหนด Octave (คู่แปด) จำนวน 6 ช่วง คู่แปด คือ Octave 1 ไปถึง Octave 6 โดยใช้แนวคิดของ อ.ขจรยศ คือใช้ O (โอ) เป็นตัวกำหนด คือจะเป็น 10, 20, 30, 40, 50, 60 โดยให้กำหนดในตอนแรกของโน้ต และช่วงที่ต้องการเปลี่ยน Octave เช่น

```
20A2,D1,F2,10G1,B2,E1,G2
```

หมายความว่า 20A2,D1,F2, เป็นเสียงใน Octave ที่ 2 เหมือนกัน 10G1,B2,E1,G2 เป็นเสียงใน Octave ที่ 1 เหมือนกัน (จำไว้ว่า ต้องกำหนด 10-60 นี้ไว้ที่หน้าตัวโน้ตตัวแรกด้วยเสมอ)

รูปแบบของไฟล์เสียง

จะประกอบด้วยส่วนสำคัญดังนี้ คือ

ก. จังหวะหรือความเร็ว กำหนดด้วยตัวเลข ตัวเลขน้อยช้า ตัวเลขมากจะไว เขียนแยกเป็นบรรทัดแรกแบบเดิม โดยผู้เขียนกำหนดให้ค่าสูงสุดคือ 400

ข. ตามด้วยการกำหนดตัวโน้ต จะมีลักษณะที่ได้กล่าวมาแล้ว เช่น

```
20A1,C#3,F2,G1,30E2,G1,A2
```

เช่น

```
200
10G1,20C2,E1,G3,G2,G1,A2,G1,10F2,A1,
G6,E3,E2,F1,E2,D1,D5,
```

มาร้องเพลงกันดีกว่า

จากแนวคิดที่ได้กล่าวมาทั้งหมด เมื่อนำมาเขียนโปรแกรมเพื่อใช้ในการเปล่งเสียงจะทำได้ดังนี้

```
200
10G1,20C2,E1,G3,G2,G1,A2,G1,F2,A1,G6,E3,E2,F1,E2,D1,D5,D1,D2
,E1,D2,C1,C6,C2,E1,G3,G2,G1,A2,G1,F2,A1,G9,G2,A1,B2,G1,30D5,
20D1,A2,B1,A6,G3,G2,A1,G2,F1,D6,F2,A1,G2,E1,C6,C2,E1,D3,D2,
D1,A3,B2,A1,G8,C1,C2,E1,G3,G2,G1,A2,B1,30C2,D1,E8,D1,C2,
20A1,G2,A1,B2,30C1,E2,C1,D2,C1,C9,
```

รูปที่ 2 ตัวอย่างการกำหนดไฟล์เสียง Thai.rpd (เพลงชาติไทย)

1. ใช้โปรแกรมพิมพ์เอกสารหรือส่วนเอดิเตอร์ของเทอร์โบปาสคาลก็ได้ เขียนส่วนของข้อมูลเสียงโดยกำหนดตัวโน้ตตามที่กล่าวมาข้างต้น และใช้เครื่องหมาย , แยกตัวโน้ตออกจากกันดังนี้

ตัวอย่างการกำหนดไฟล์เสียง thai.rpd (เพลงชาติไทย) จะเป็นดังรูปที่ 2

เมื่อเรียบร้อยให้เก็บลงในชื่อ thai.rpd (ส่วนขยายจะเปลี่ยนเป็นชื่ออื่นก็ได้)

2. เขียนโปรแกรมบนเทอร์โบ-ปาสคาล เพื่ออ่านข้อมูลเสียงที่กำหนดไว้

นี้มารัน โดยจะมีแนวคิดดังไฟล์ song.pas ซึ่งขอแยกอธิบายเป็น 2 ส่วน โดยในช่วงนี้ขอให้ศึกษาโปรแกรมในรูปที่ 3 ก่อน มีรายละเอียดดังนี้

1. บรรทัดที่ 3-7 กำหนดลักษณะของข้อมูลเสียง ประกอบด้วย ตัวโน้ต ครึ่งเสียง ช่วงคู่แปด และค่าหน่วงเวลาในการเปล่งเสียง

2. บรรทัดที่ 8-14 กำหนดความถี่เสียงตามตัวโน้ตในแต่ละช่วงคู่แปด จำนวน 6 ช่วงคู่แปด โดยให้ความถี่สุดท้ายมีค่า 30,000 เฮิรตซ์ ซึ่งเป็นเสียงที่คนเราไม่ได้ยิน (เราได้ยินเสียงในช่วง 20-20,000 เฮิรตซ์)

3. บรรทัดที่ 15-18 กำหนดตัวแปรใช้งาน

4. บรรทัดที่ 19-37 ส่วนการเปล่งเสียงตามตัวโน้ตที่กำหนด

5. บรรทัดที่ 22-32 คำนวณหาระดับเสียงที่

6. บรรทัดที่ 33-36 ส่วนการเปล่งเสียงตามระดับเสียงที่ช่วงคู่แปดและระดับเสียงที่คำนวณได้

เมื่อเราได้โปรแกรมในการนำตัวโน้ตมาเปล่งเสียงตามความต้องการได้แล้ว ในช่วงต่อไปก็คือการเขียนโปรแกรมอ่านข้อมูลเสียงที่เก็บไว้ในแผ่นดิสก์เข้ามาเพื่อให้คอมพิวเตอร์ร้องเพลง

โดยผู้เขียนใช้แนวทางในการเขียน

```
 1: PROGRAM Song;
 2: uses crt;
 3: Type NoteData=record
 4:                 Note,Shape:char;
 5:                 Octave:byte;
 6:                 time:word;
 7:               end;
 8: const Frequency:array[1..6,1..13] of word=
 9:   ((131,139,147,156,165,175,185,196,208,220,233,247,30000),
10:    (262,277,294,311,330,349,370,392,415,440,466,494,30000),
11:    (523,554,587,622,659,698,740,784,831,880,932,988,30000),
12:    (1047,1109,1175,1245,1319,1397,1480,1568,1661,1760,1865,1976,30000),
13:    (2093,2217,2349,2489,2637,2794,2960,3136,3322,3520,3729,3951,30000),
14:    (4186,4435,4699,4978,5274,5587,5919,6271,6645,7040,7459,7902,30000));
15: var Round:integer;
16:     TimeSong,Maxsong,R:word;
17:     Note:array[1..193] of NoteData;
18:     Path:string;

19: PROCEDURE Ssong(No:byte);
20: var Level:byte;
21: begin
22:    Level:=13;
23:   Case Note[no].note of
24:     'C':Level:=1;
25:     'D':Level:=3;
26:     'E':Level:=5;
27:     'F':Level:=6;
28:     'G':Level:=8;
29:     'A':Level:=10;
30:     'B':Level:=12;
31:     'P':Level:=13;
32:  end;
33:    nosound; delay(5);
34:    if (Note[no].shape='#') and (Level in [1,3,6,8,10]) then Inc(Level);
35:    sound(Frequency[Note[no].Octave, Level]);
36:    delay(Note[no].Time*Timesong-5);
37: end;
```

รูปที่ 3
ส่วนการกำหนดความถี่เสียง ตัวแปรที่เกี่ยวข้อง และส่วนการเปล่งเสียง

โปรแกรมดังนี้

1. กำหนดให้ไฟล์ข้อมูลเสียงยาวไม่เกิน 2,000 ไบต์

2. โปรแกรมจะตรวจรหัสแอสกีเพียงบางตัวเท่านั้น คือจะตรวจ A-G, P, O, #, 0-9 โดยจะถือว่าตัวเล็กหรือใหญ่เป็นตัวเดียวกัน

จำนวนข้อมูลเสียงจะขึ้นกับเครื่องหมาย , ที่กำหนดไว้

4. โปรแกรมจะเริ่มต้นด้วยการตรวจจังหวะหรือความเร็วก่อน และตามด้วยการตรวจตัวโน้ตของเสียงเพลงที่กำหนด

5. เมื่อการตรวจสิ้นสุด จึงจะนำข้อมูลทั้งหมดไปเปล่งเสียงออกทางลำโพงต่อไป

เมื่อนำมาเขียนโปรแกรมจะได้ดังรูปที่ 4 โดยมีรายละเอียดดังนี้

1. บรรทัดที่ 45-47 ตรวจสอบว่ามีไฟล์ข้อมูลเสียงตามที่ระบุชื่อในแผ่นดิสค์หรือไม่ ถ้ามีจะทำงานในบรรทัดที่ 47-104 ถ้าไม่มีก็จะแสดงข้อความให้ทราบตามบรรทัดที่ 106

2. บรรทัดที่ 55-62 ตรวจสอบหาจังหวะหรือความไว โดยจะนำค่าที่ตรวจได้มาลบออกจาก 400 มาเก็บไว้ในตัวแปร TimeSong

3. บรรทัดที่ 64-100 ช่วงวนรอบตรวจหาข้อมูลต่าง ๆ ของตัวโน้ต

4. บรรทัดที่ 70-71 ตรวจหาตัวโน้ตจากการกำหนด A-G และ P

5. บรรทัดที่ 72 ตรวจว่าเป็นครึ่งเสียงหรือไม่

6. บรรทัดที่ 73-85 ตรวจหาช่วงเวลาของการเปล่งเสียง (0-9) และช่วงคู่แปด (0, 0-6)

7. บรรทัดที่ 86-92 ตรวจการแบ่งตัวโน้ตของการส่วน ๆ ด้วยเครื่องหมาย ,

รูปที่ 4
ชุดคำสั่งอ่านข้อมูลเสียง
จากไฟล์ในแผ่นดิสก์
เพื่อนำมาเปล่งเสียง

```
38: PROCEDURE MainSong(FileSong:string);
39: var k,size,No,count:word;
40:     f:file;  ck:integer;
41:     Data:array[1..2000] of char;
42:     ST,TimeST:string;
43:     Octave:byte;
44: begin
45:   assign(F,FileSong);
46:  {$I-}   reset(f,1); {$I+}
47:  if IoResult=0 then
48:   begin
49:      writeln('song of ',filesong,#13#10);
50:      write(Path,'>');
51:      size:=filesize(f);
52:      if size>2000 then Size:=2000;
53:      blockread(f,Data,size);
54:      close(f);
55:      k:=1; ST:='';
56:       repeat
57:         if Data[k] in ['0'..'9'] then ST:=ST+Data[k];
58:         inc(k);
59:       until Data[k] in [#10,#13];
60:        val(ST,TimeSong,ck);
61:        if Timesong>399 then TimeSong:=399;
62:        TimeSong:=400-Timesong;
63:        No:=1; MaxSong:=0;  Octave:=1;
64:       repeat
65:        Note[No].Note:=#0;  Note[no].Shape:=#0;
66:        Note[no].Octave:=Octave; Note[no].time:=27;  TimeST:='';
67:        repeat
68:           inc(k);
69:         case Upcase(Data[k]) of
70:           'P','A'..'G':if Note[no].Note=#0 then
71:                          Note[no].Note:=Upcase(Data[k]);
72:           '#':Note[no].Shape:='#';
73:           '0'..'9':begin
74:                      if Upcase(Data[k+1])='O' then
75:                       begin
76:                         if Data[k] in ['1'..'6'] then
77:                          begin
78:                            Note[no].Octave:=Ord(Data[k])-48;
79:                            Octave:=Note[no].Octave;
80:                            Inc(k);
81:                          end;
82:                       end
83:                       else
84:                         TimeST:=TimeST+Data[k];
85:                    end;
86:           ',':if Note[no].note<>#0 then inc(MaxSong)
87:                    else
88:                    begin
89:                       Note[No].Note:=#0;  Note[no].Shape:=#0;
90:                       Note[no].Octave:=Octave; Note[no].time:=27;
91:                       TimeST:='';
92:                    end;
93:         end;
94:        until Data[k] in [',',#26];
95:           val(Timest,Note[no].time,ck);
96:           if Note[no].Note=#0 then Note[no].Note:='P';
97:           inc(No);
98:           if Data[k]=#26 then k:=Size+2:
```

รูปที่ 4 (ต่อ)

```
 99:            if Maxsong>193 then k:=Size+2;
100:        until k>size;
101:         for k:=1 to maxSong do Ssong(k);
102:        Nosound;
103:        gotoxy(1,whereY-1);
104:      end
105:    else
106:       writeln('file '#39,Filesong,#39' not found'#13#10);
107: end;

108: {================== SONG.PAS ====================}
109: begin
110: clrscr;  nosound;
111: writeln('=== SONG.EXE ========== THAISHOW 3.0 ==========');
112: writeln('  By  Arthan Sattayarak lampang Tel. 054-223091'#13#10);
113: getdir(0,path);
114: round:=1; if Paramstr(2)<>'' then val(Paramstr(2),Round,Round);
115: if Paramstr(1)<>'' then Mainsong(Paramstr(1))
116:  else
117:   begin
118:     writeln('usage : song  fileofsong');
119:   end;
120: end.
```

8. บรรทัดที่ 101 นำข้อมูลตัวโน้ตที่ได้ไปเปล่งเสียงเป็นเพลง

9. บรรทัดที่ 108-120 ส่วนของโปรแกรมหลัก ซึ่งได้กำหนดให้ป้อนชื่อไฟล์นี้ ตามด้วยชื่อไฟล์ข้อมูลเสียง

เมื่อคอมพายล์ลงแผ่นดิสก์จะได้ไฟล์ชื่อ song.exe นำไปใช้งานได้ ดังนี้

```
A:\ >song thai.rpd
```

โดยให้ป้อนชื่อไฟล์ข้อมูลเสียงต่อท้าย ถ้าหากผู้อ่านจะนำบางส่วนของไฟล์ song.pas นี้ไปใช้งานต่อไป ขอให้ผู้อ่านทำดังนี้

1. ใช้โปรแกรมตั้งแต่บรรทัดที่ 1 ไปจนถึงบรรทัดที่ 107 เท่านั้น บรรทัดอื่น ๆ ท้ายนี้ให้ตัดทิ้งไป

2. ตัดบรรทัดที่ 49, 50, 101, 105 และ 106 ออกไป

3. เปลี่ยนบรรทัดที่ 104 จาก end เป็น end;

4. ทำการเก็บไฟล์ที่ได้ทำตามข้อ 1-3 นี้ลงในแผ่นดิสก์ในชื่อ head.pas

การวนรอบร้องเพลง

ไฟล์ head.pas นี้ผู้เขียนขอนำไปใช้ในไฟล์ song2.pas เพื่อเสนอแนวทางการเปล่งเสียงเพลงในช่วงของการวนรอบรอการกดแป้น เช่น ในช่วงที่เราแสดงเมนูรายการเลือกต่าง ๆ ซึ่งจะช่วยเพิ่มสีสันให้กับโปรแกรมได้อีกรูปแบบหนึ่ง

โดยโปรแกรมจะเป็นดังรูปที่ 5 และมีรายละเอียดของไฟล์ song2.pas ดังนี้

รูปที่ 5 ชุดคำสั่งการกำหนดข้อมูลเสียงลงในไฟล์ และการวนรอบการเปล่งเสียงเพลง

```
1: {$I Head}

2: PROCEDURE SongForYou(Var NoSong:byte);
3: begin
4:    Ssong(Nosong);
5:    Inc(Nosong);
6:    if NoSong>Maxsong then Nosong:=1;
7: end;

8: PROCEDURE PinkSong;
9: const PinkSong:array[1..75] of char=
10:   ('G','C','E','G','G','G','G','G','G','G',
11:    'G','G','F','E','F','E','E','E','D','F',
12:    'F','F','F','F','F','F','F','F','G','A',
13:    'B','A','G','F','E','G','G','G','G','G',
14:    'G','G','G','G','G','C','E','G','F','E',
15:    'F','E','E','D','A','A','A','A','A','A',
16:    'G','G','E','E','E','D','C','C','C','C',
```

รูปที่ 5 (ต่อ)

```
    'C','C','C','C','C');
var k:byte;
begin
  TimeSong:=200;  Maxsong:=75;
  for k:=1 to 75 do
   begin
     case k of
        1:Note[k].Octave:=2;
        2..44:Note[k].Octave:=2;
         45:Note[k].Octave:=2;
       46..75:Note[k].Octave:=3;
     end;
    Note[k].Shape:=#0;
    Note[k].Note:=Pinksong[k];
    Note[k].time:=1;
   end;
end;

{============== SONG2.PAS ==================}
var No:byte; Ch:char;
begin
  clrscr;
  PinkSong;
  writeln('song from PROCEDURE PinkSong;');
  No:=1;
 repeat
    SongForYou(No);
    delay(20);
 until keypressed;
  writeln('song of thai.rpd');
  MainSong('thai.rpd');
  No:=1; ch:=readkey;
  repeat
    SongForYou(No);
  until keypressed;
  Nosound;
end.
```

■

1. บรรทัดที่ 1 เรียกไฟล์ head.pas ตามที่กล่าวมาแล้วมาใช้งานด้วย

2. บรรทัดที่ 2-7 ส่วนการวนรอบเพื่อเปล่งเสียงเพลง

3. บรรทัดที่ 8-33 ตัวอย่างการนำข้อมูลเสียงมากำหนดไว้ในไฟล์ *.pas โดยบรรทัดที่ 23-28 จะเป็นส่วนการกำหนดช่วงคู่แปด หากไฟล์เสียงของท่านมีคลื่นเสียง ก็ต้องกำหนดให้กับโปรแกรมด้วย ซึ่งคงไม่ยากเกินกว่าที่ท่านจะทำได้ด้วยตนเอง (สำหรับเพลงนี้คือเพลงอะไรคงต้องฟังจากการเล่นของโปรแกรม)

4. บรรทัดที่ 38 กำหนดข้อมูลเสียงตามชุดคำสั่ง PinkSong

5. บรรทัดที่ 40-44 วนรอบเล่นเพลงนี้จนกว่าจะมีการกดแป้น

6. บรรทัดที่ 46 อ่านข้อมูลไฟล์เสียง thai.rpd มาจากแผ่นดิสค์

7. บรรทัดที่ 47-51 วนรอบร้องเพลงชาติไทย จนกว่าจะกดแป้น

หวังว่าผู้อ่านคงได้แนวทางในการทำให้คอมพิวเตอร์ร้องเพลงอย่างสมบูรณ์แล้ว การใช้งานจริงจึงอยู่ที่ความคิดของผู้อ่านเป็นสำคัญ

ก่อนจาก

ก่อนจบบทความชุดนี้ ผู้เขียนขอฝากข่าวถึงผู้ใช้โปรแกรมกราฟิก THAISHOW 3.0 ดังนี้

1. ผู้เขียนได้ปรับปรุงการเปล่งเสียงจากที่กำหนดเดิมมาเป็นการเปล่งเสียงตามข้อมูลเสียงที่กำหนดใหม่นี้เรียบร้อยแล้ว

2. ผลการปรับปรุงจึงทำให้ไฟล์ THAISHOW.EXE, SONG.EXE มีการเปลี่ยนแปลงคือ สามารถใช้ข้อมูลเสียงตามแนวคิดเดิมและข้อมูลเสียงตามแนวคิดใหม่ได้

ถ้าผู้อ่านต้องการไฟล์ทั้งสองนี้เพื่อนำไปใช้งาน ท่านสามารถทำได้ 2 วิธีคือ

วิธีแรก ส่งแผ่นดิสค์โปรแกรมกราฟิก THAISHOW 3.0 (ต้นฉบับ) พร้อมซองเขียนชื่อที่อยู่ของท่าน และติดแสตมป์ 7 บาท (เพื่อบรรจุแผ่นดิสค์ส่งกลับคืนให้ท่านทางพัสดุไปรษณีย์)

วิธีสอง ส่งเงินจำนวน 150 บาท เพื่อสั่งซื้อแผ่นเอดิเตอร์ของ THAISHOW 3.0 ซึ่งท่านจะได้ไฟล์ editor.exe, pic.exe, font.exe, thaishow.exe, song.exe และไฟล์ข้อมูลเสียงอีก 6 ไฟล์ ซึ่งในแผ่นดิสค์จะมีคำอธิบายการใช้โปรแกรมเอดิเตอร์มาให้ด้วย โดยส่งไปที่ ตู้ ปณ.34 อ.เมือง จ.ลำปาง 52000 สำหรับฉบับนี้ ขอจบเพียงแค่นี้ครับ...